Introduction to Robotics for cognitive science

Dr. Andrej Lúčny

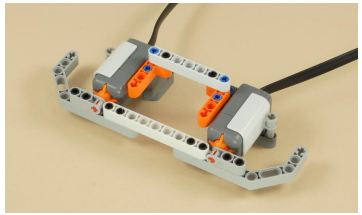
KAI FMFI UK

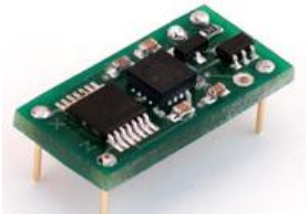
lucny@fmph.uniba.sk

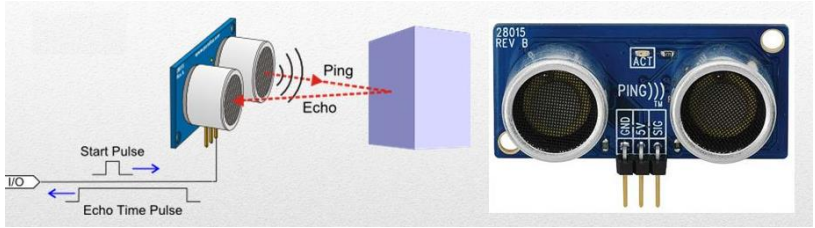
Web page of the subject

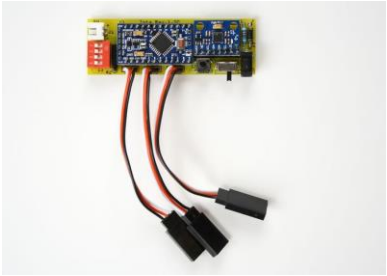
www.agentspace.org/kv

Sensors

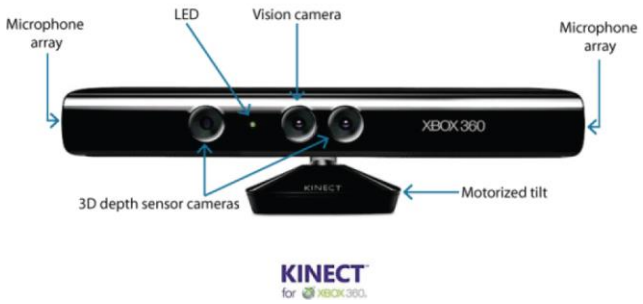





Ping
Echo
Start Pulse
I/O
Echo Time Pulse
28015
REV B
PING)))
GND
5V
SIG




Velodyne


Microphone array
LED
Vision camera
Microphone array
XBOX 360
KINECT
3D depth sensor cameras
Motorized tilt
KINECT
for XBOX 360

Sensors

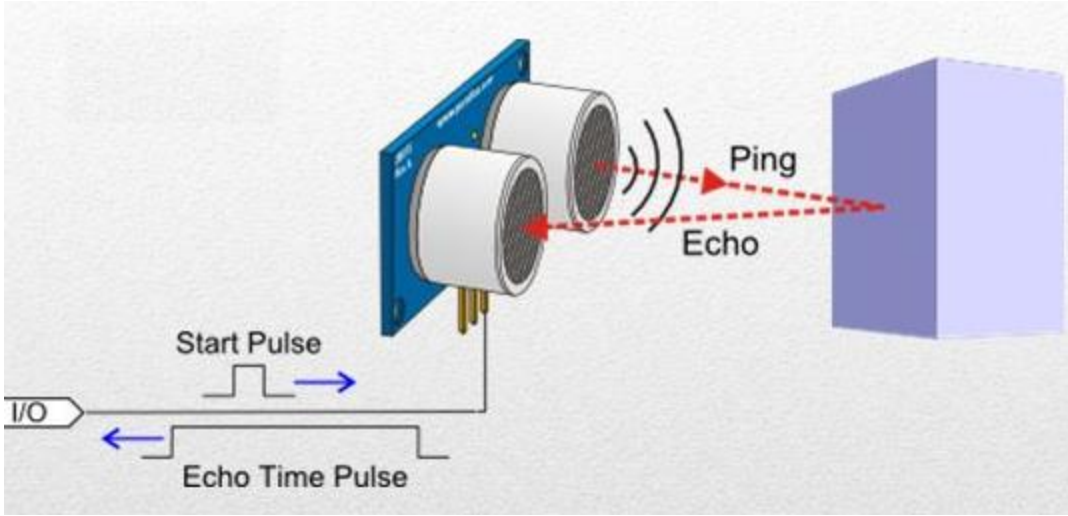
- Bumper (0/1)

Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)

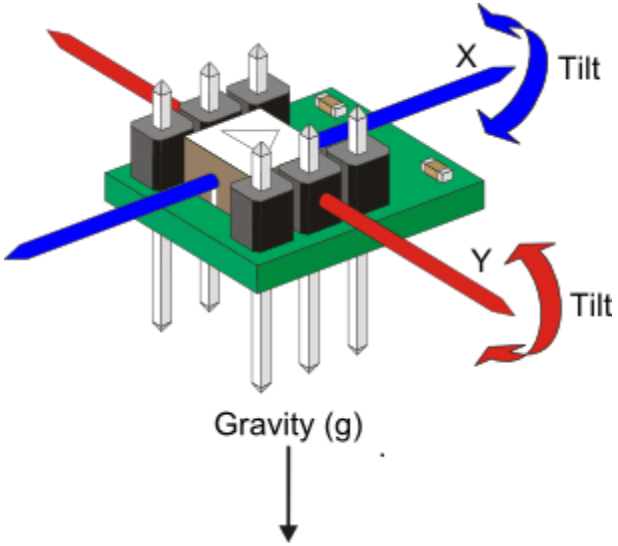
Sensors

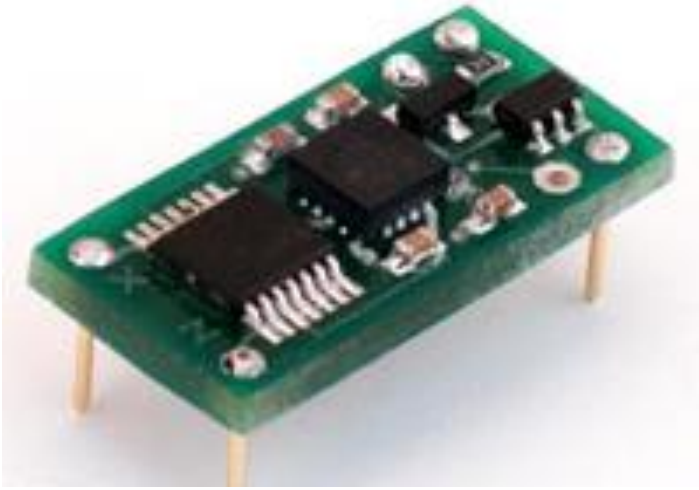
- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)



Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (accelarations)



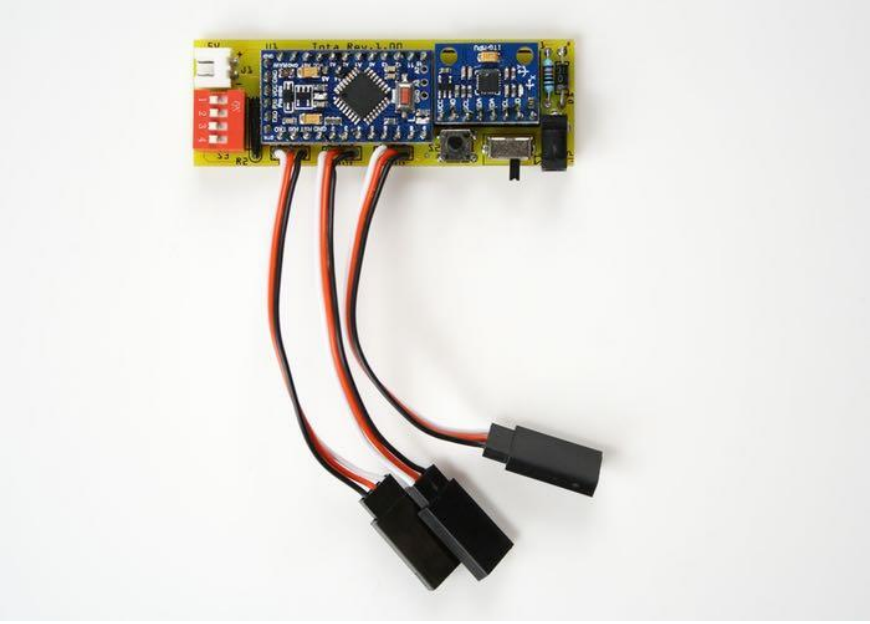


Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (accelerations)
- Tachometer (rpm = revolutions)

Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (accelerations)
- Tachometer (rpm = revolutions)
- Gyroscope (angular velocity)



Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (accelerations)
- Tachometer (rpm = revolutions)
- Gyroscope (angular velocity)
- Camera (image)

Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer
- Tachometer (rp
- Gyroscope (ang
- Camera (image
- 3D camera (image + depth)

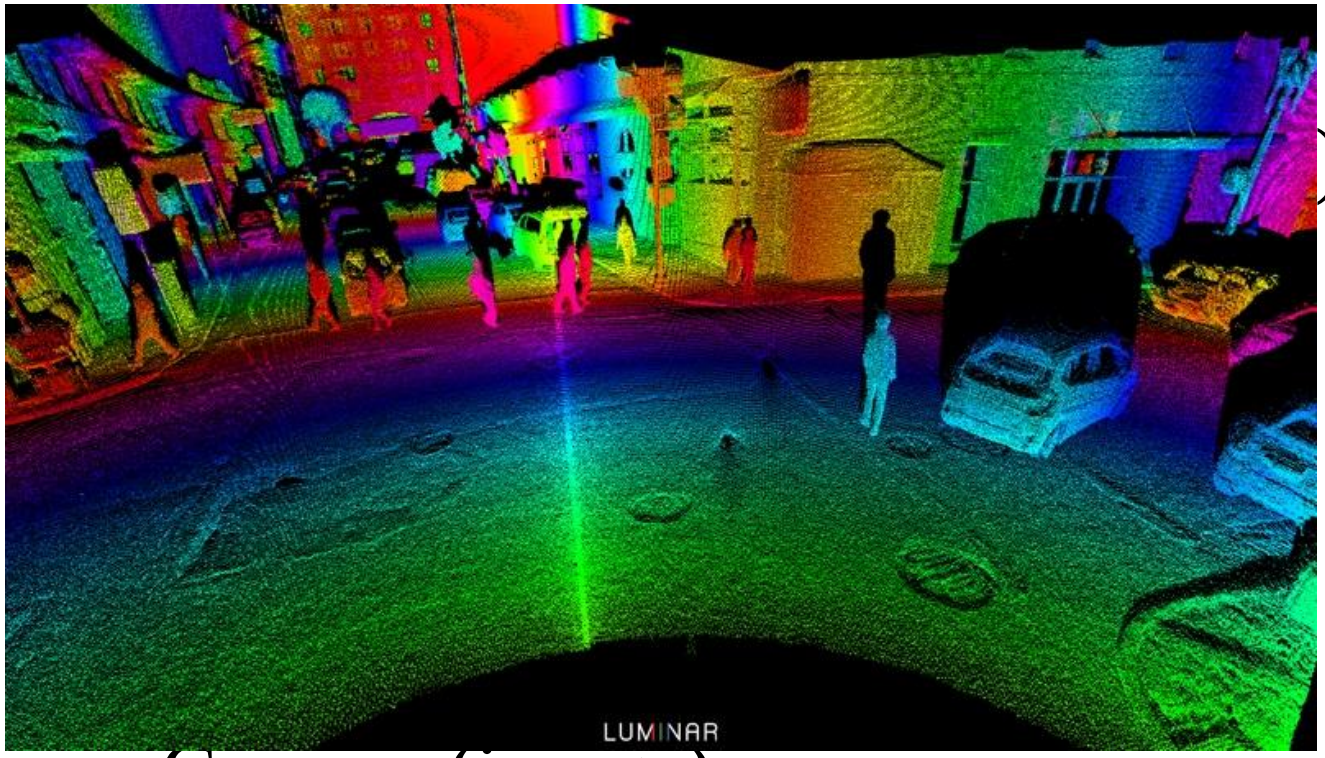
Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (a
- Tachometer (rpm
- Gyroscope (angu
- Camera (image)
- 3D camera (imag
- Laser profile sensor (distance 2D, 3D)

Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (accelerations)
- Tachometer (rpm = revolutions)
- Gyroscope (angular velo
- Camera (image)
- 3D camera (image + dep
- Laser profile sensor (dist
- doppler radar (depth map)

Sensors



- Camera (image)
- 3D camera (image + depth)
- Laser profile sensor (distance 2D, 3
- doppler radar (depth map)
- LIDAR (depth map)

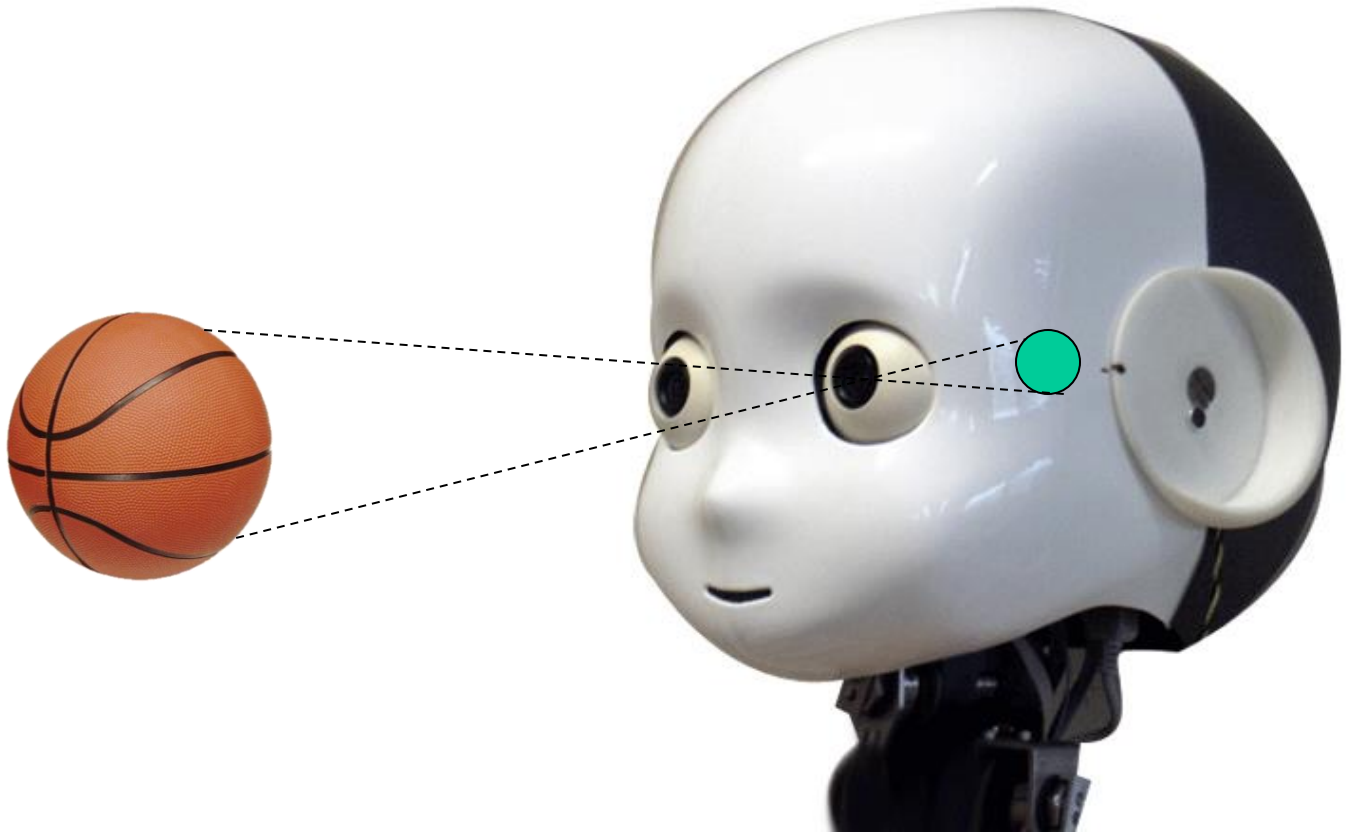
Sensors

- Bumper (0/1)
- Infra red sensor (distance or color)
- Sonar (distance)
- Accelerometer (accelerations)
- Tachometer (rpm = revolutions)
- Gyroscope (angular velocity)
- Camera (image)
- 3D camera (image + depth)
- Laser profile sensor (distance 2D, 3D)
- doppler radar (depth map)
- LIDAR (depth map)

Sensor parameters

- Range
- Accuracy
- Resolution
- FOV (field of view)
- Fps
- Supply voltage
- Power consumption
- Interface (TTL, RS232, RS485, USB, Ethernet, …)

Perception



iCub 2007

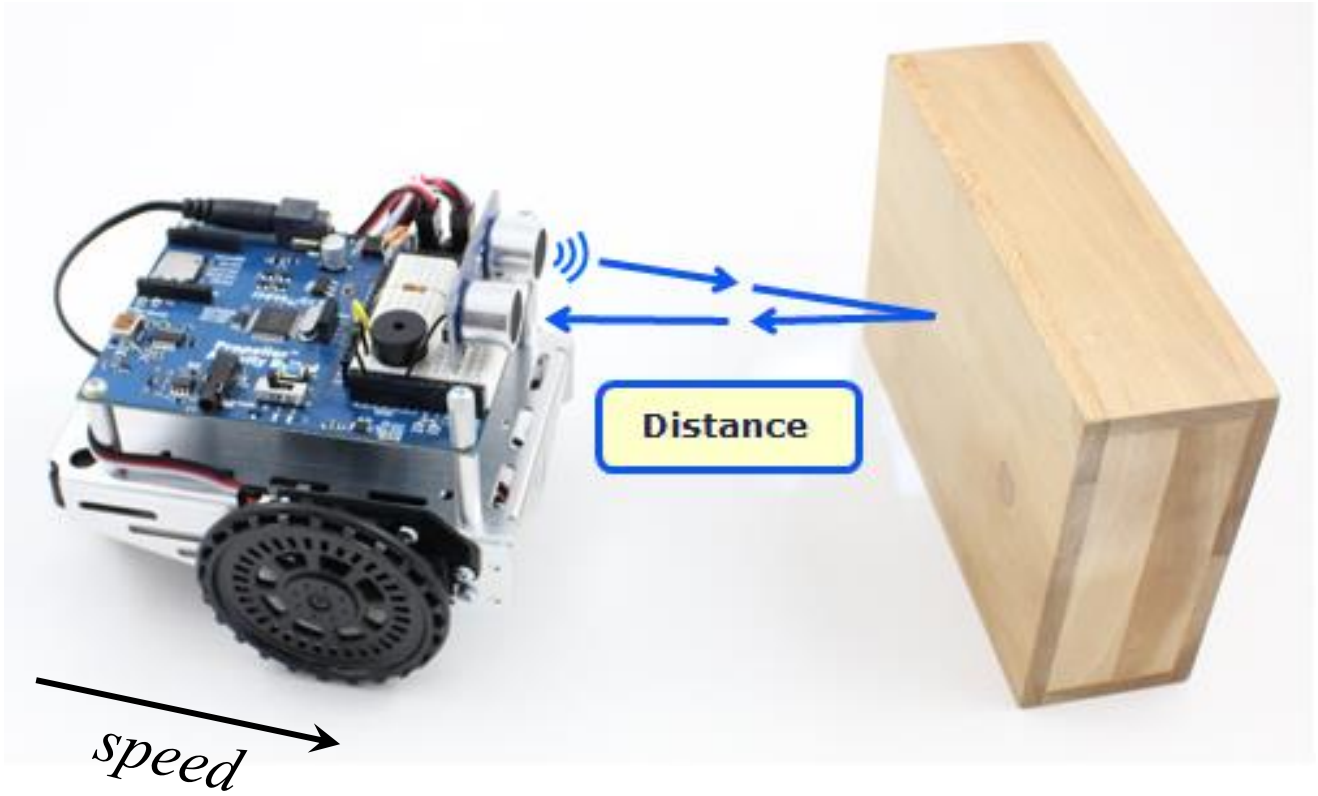
Sensorimotor approach

- usual solution: stationary perception
- sensory-motor approach: constitute actions to improve perception
- Examples:
  - intense light causes the closing of eyelids
  - dark causes stretch of pupils
  - when recognition fails, we move our head
  - we treat convolution caused by body movement by contra movement of the extraocular muscles

Processing one value

- Processing of measured distance, speed, acceleration, … is relatively easy
- We need to consider just that any measurement has an accuracy
- So even when the robot stops, the measured distance to close obstacle varies and oscillates
- If we know how to predict the future state of our robot from the current state, the variation can be decreased by the Kalman filter

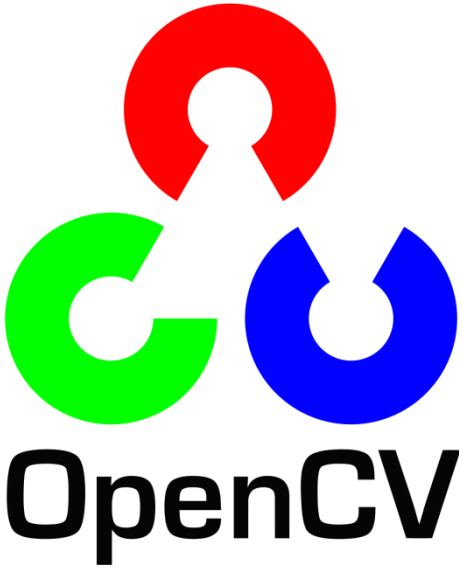
Robot moving to obstacle



- When a robot measures the distance to an obstacle, it has an inner state – speed, which we also know just with some accuracy and which can be derived from distance measurement
- from the speed, we can predict what distance will be measured in the next instant of time
- In the next instant of time, we have two values – measured and predicted
- Kalman filter tells us the best estimation of the actual distance value under these uncertain circumstances and derives the best estimation for the next value of the inner state

Computer vision

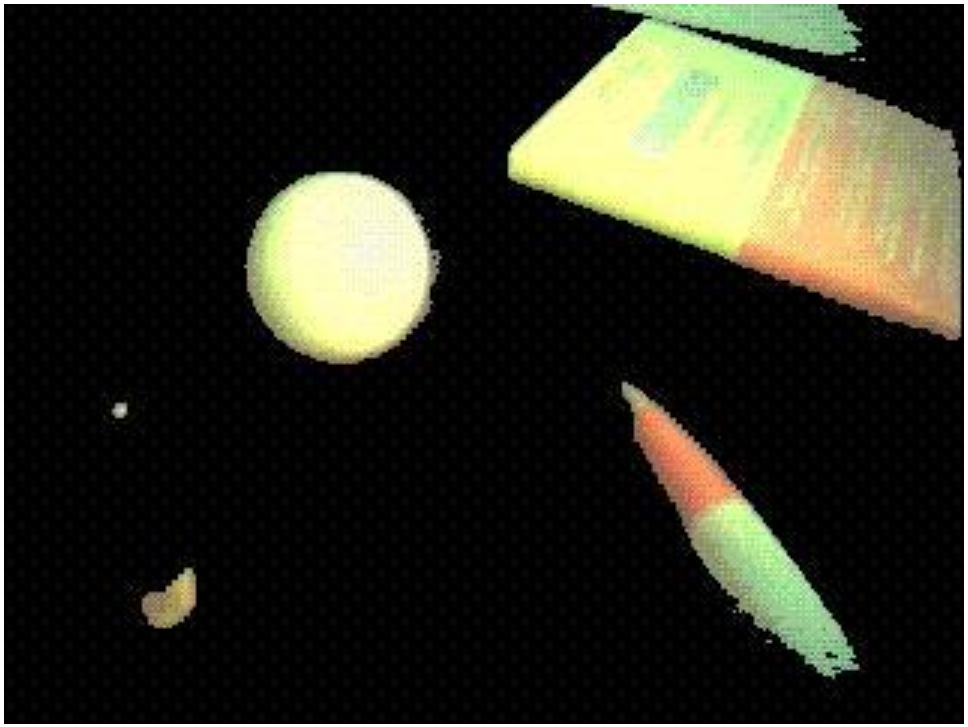
- Much more complicated is the processing of image or depth map provided by camera, radar, or LIDAR
- Typical processes:
  - Filtering
  - Shape detection
  - Object detection
  - Object tracking
  - Segmentation



- Open source library for computer vision
- Intel, Willow Garage, Itseez
- BSD License
- > 2500 algorithms
- C++, Java, Python
- Windows, Linux, Android
- fast & reliable
- CUDA & OpenCL support

```
pip install opencv-contrib-python
```

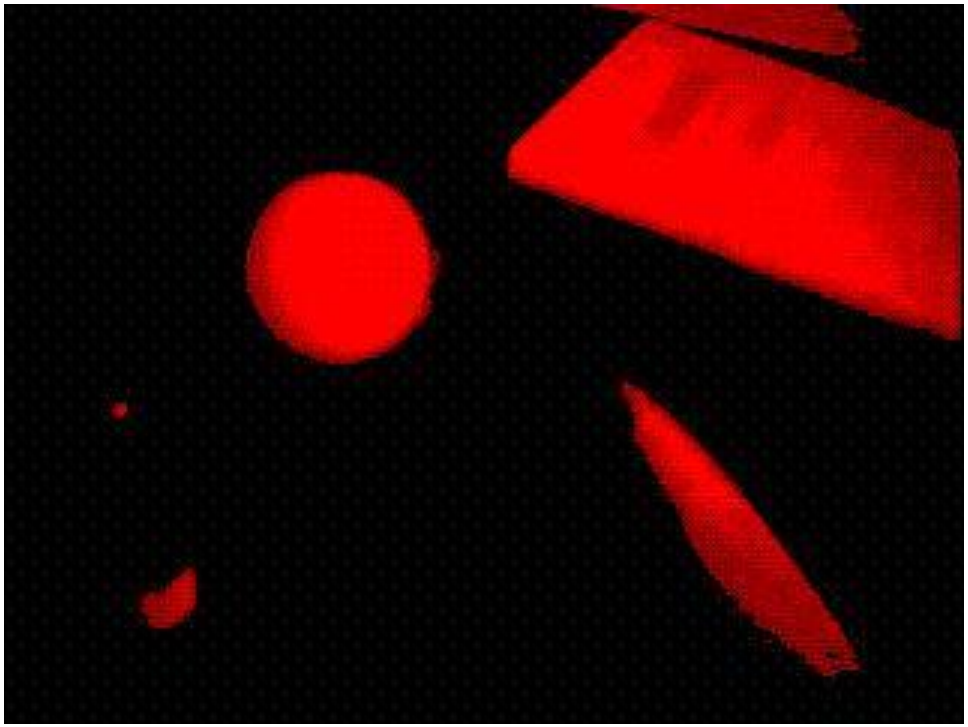
Camera provides image



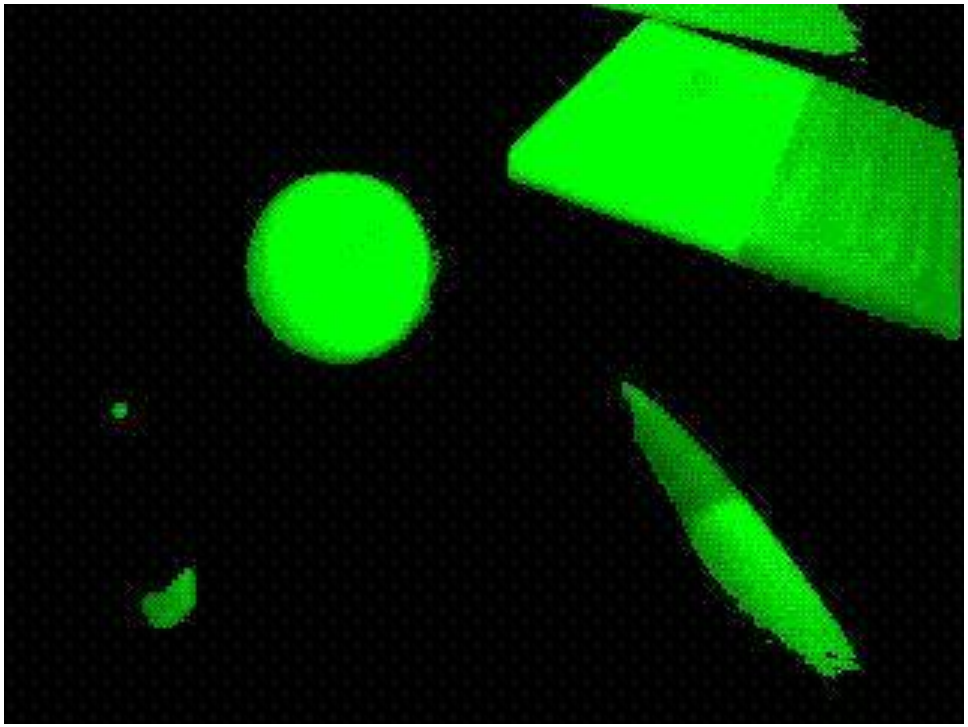
numpy array with shape (rows, columns, channels)

index [0,0,;] or point (0,0) is top left

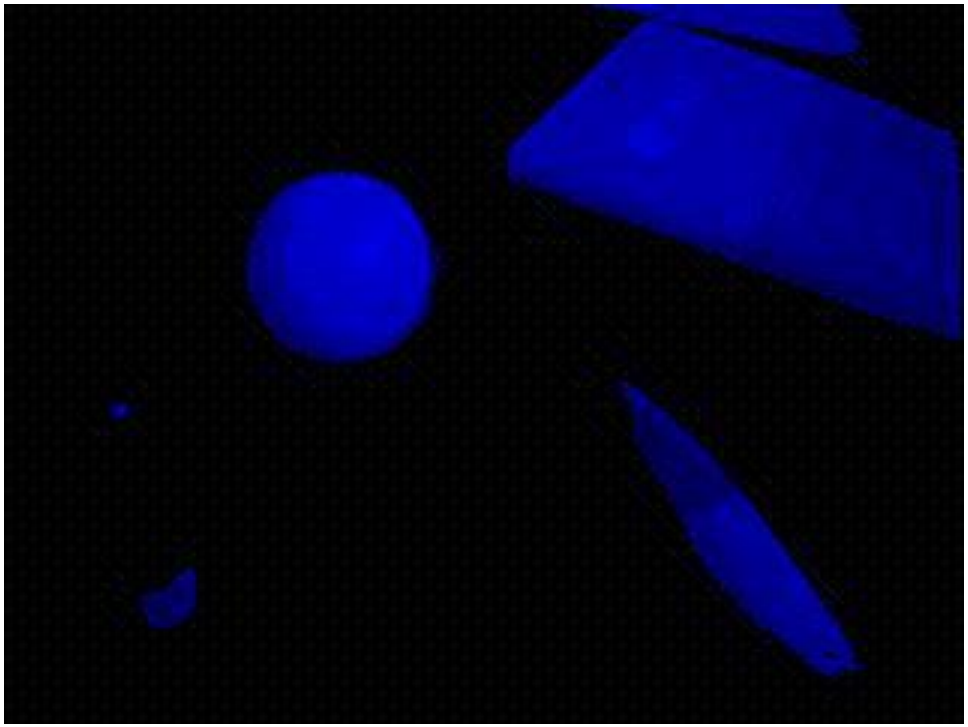
Red channel



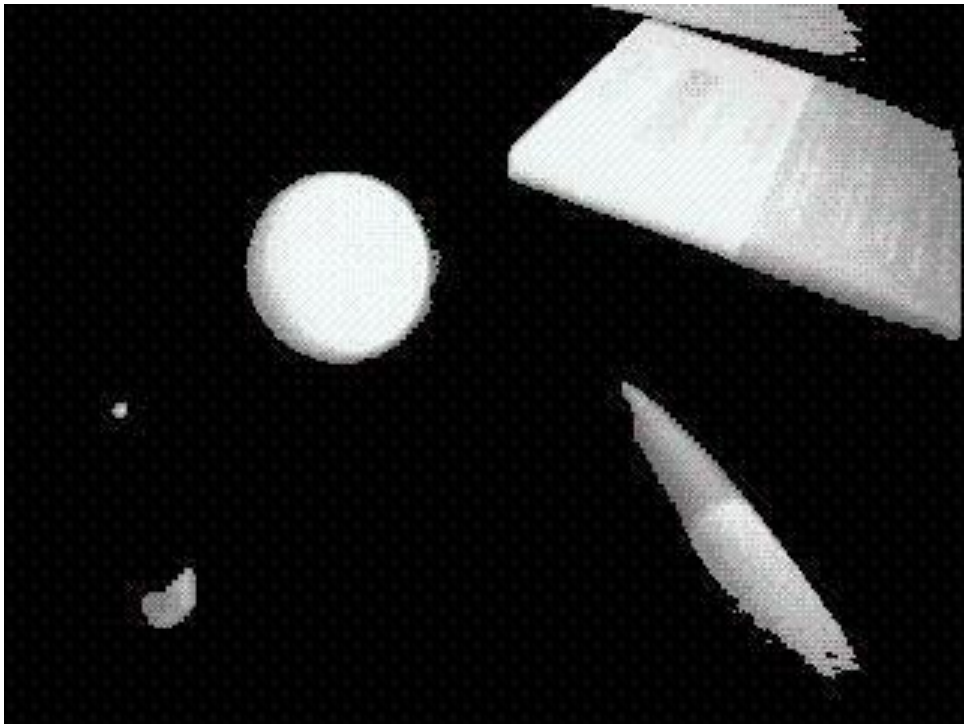
Green channel



Blue channel



Grayscale image

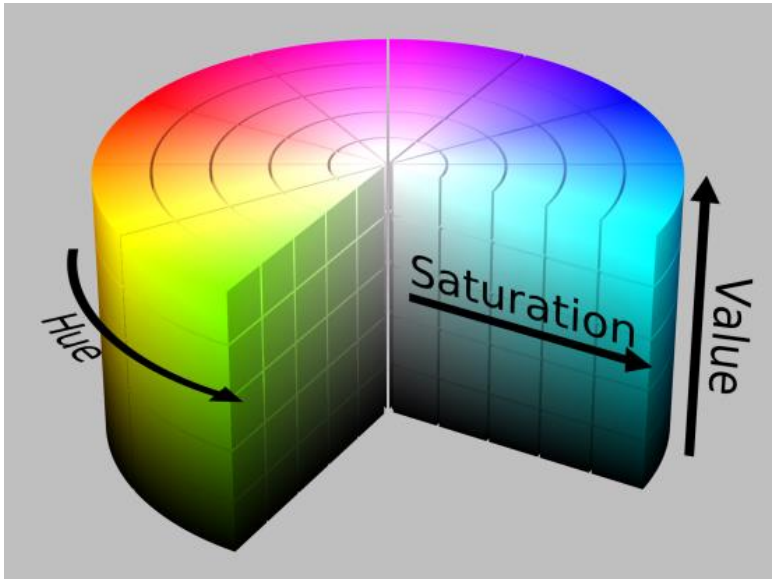


$$\text{grey} = 0.3 \times \text{red} + 0.59 \times \text{green} + 0.11 \times \text{blue}$$

Color models

- RGB used by cameras is just one of the possible color model
- OpenCV employs mainly BGR model., `(255,0,0)` is blue, `Scalar(0,0,255)` is read, `Scalar(255,0,255)` is magenta, `Scalar(127,127,127)` grey, ...

- OpenCV can convert one model to another, e.g., to HSV, which is suitable for color filtering
- Other models: YUV, YCrCb, HLS, CIE 1976 L*a*b*



Brightness

+ bias =

Contrast

× weight =

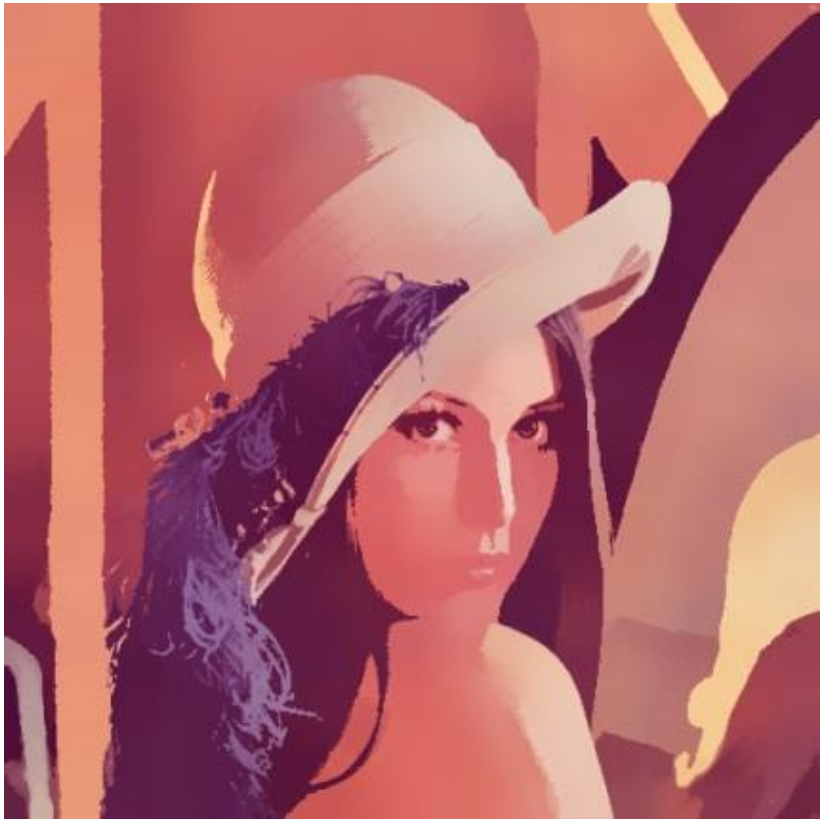
Filtering: blur

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

```
dst = cv2.blur(src,(kx,ky))
```

Many others: median filter, bilateral filter, anisotropic diffusion …

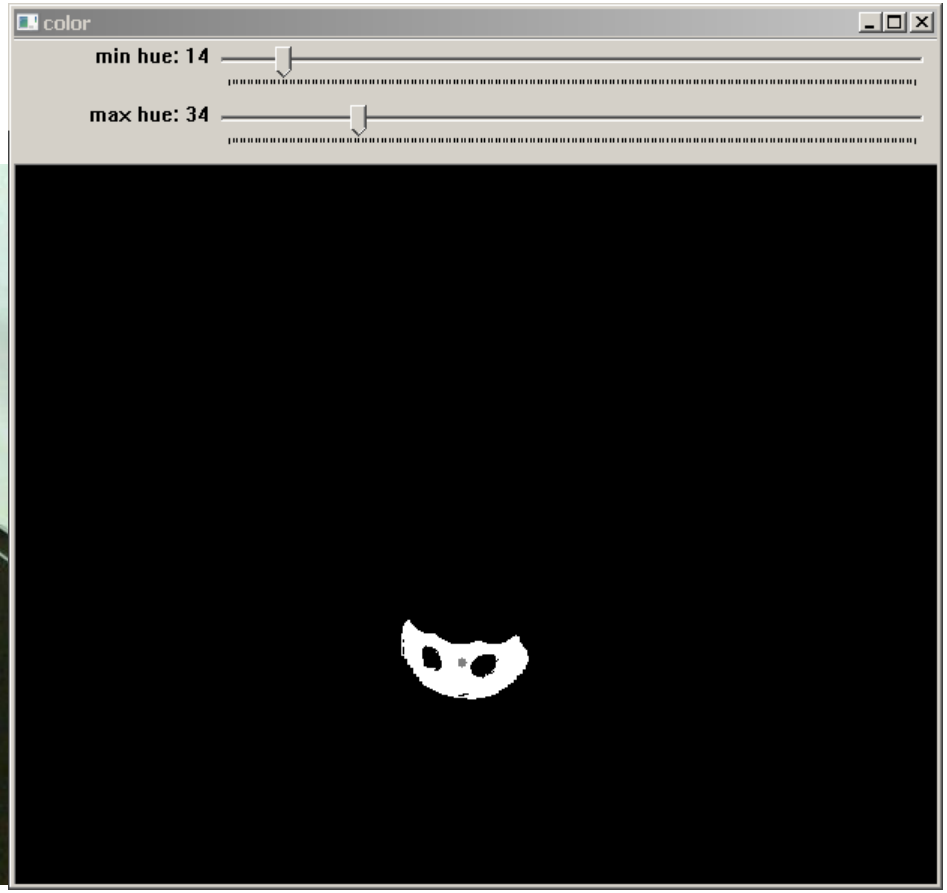
Filtering: meanshift



```
img_filtered = cv2.pyrMeanShiftFiltering(
                    img, spatial_radius, color_radius)
```

Filtering: color detection

yellow color in HSV

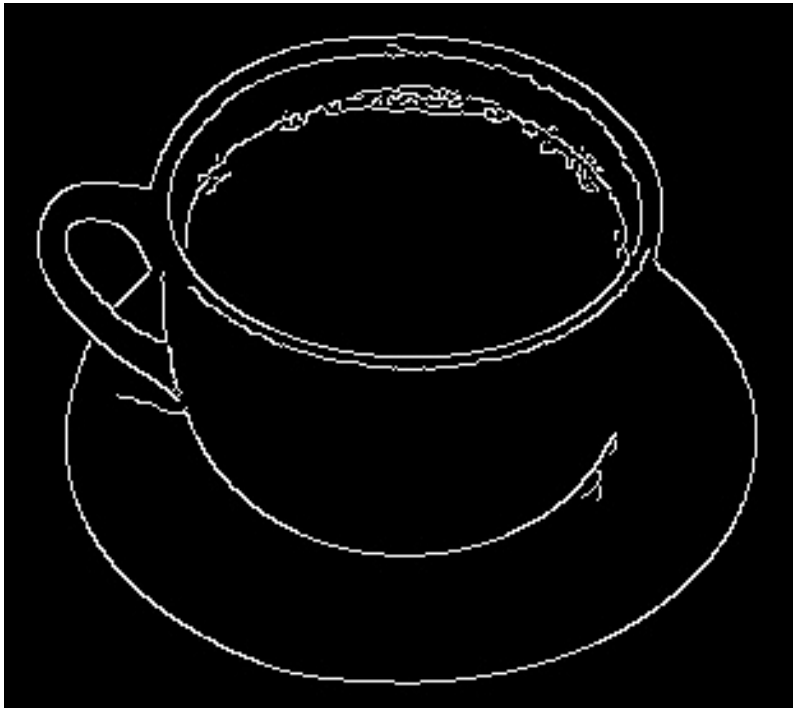


Filtering: edge detection (Sobel)

horizontal Sobel kernel:

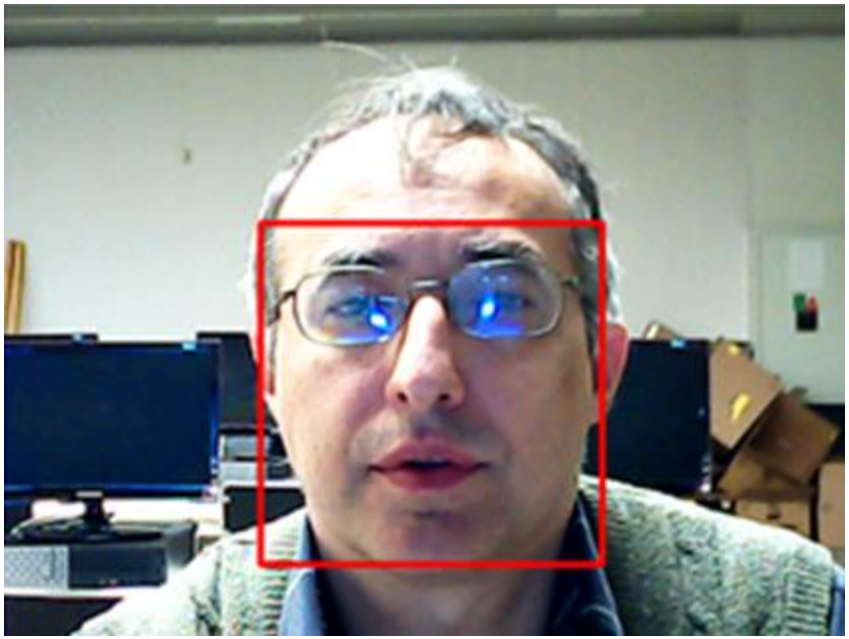
-1	-2	-1
0	0	0
1	2	1

Filtering: edge detection (Canny)

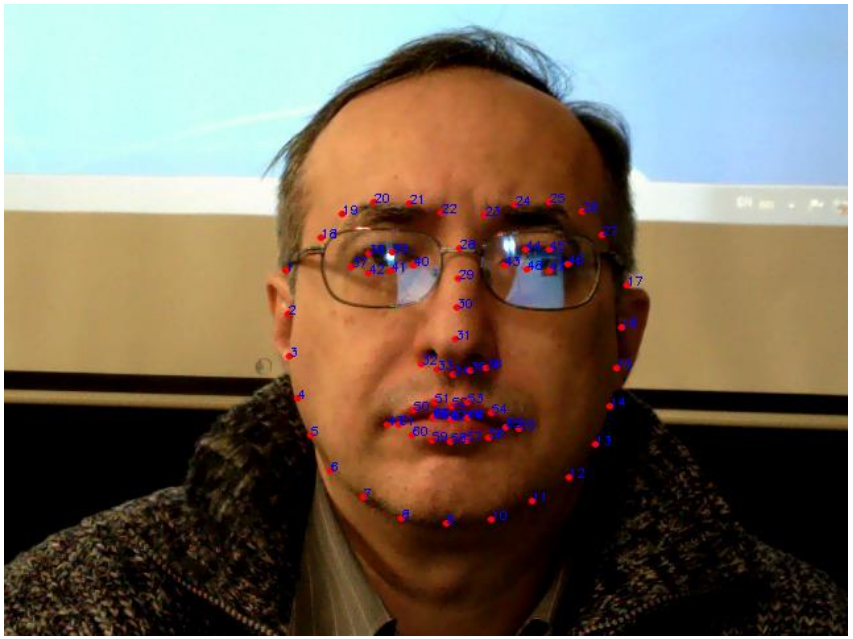


Shape detection

Face detection,
Face landmark detection

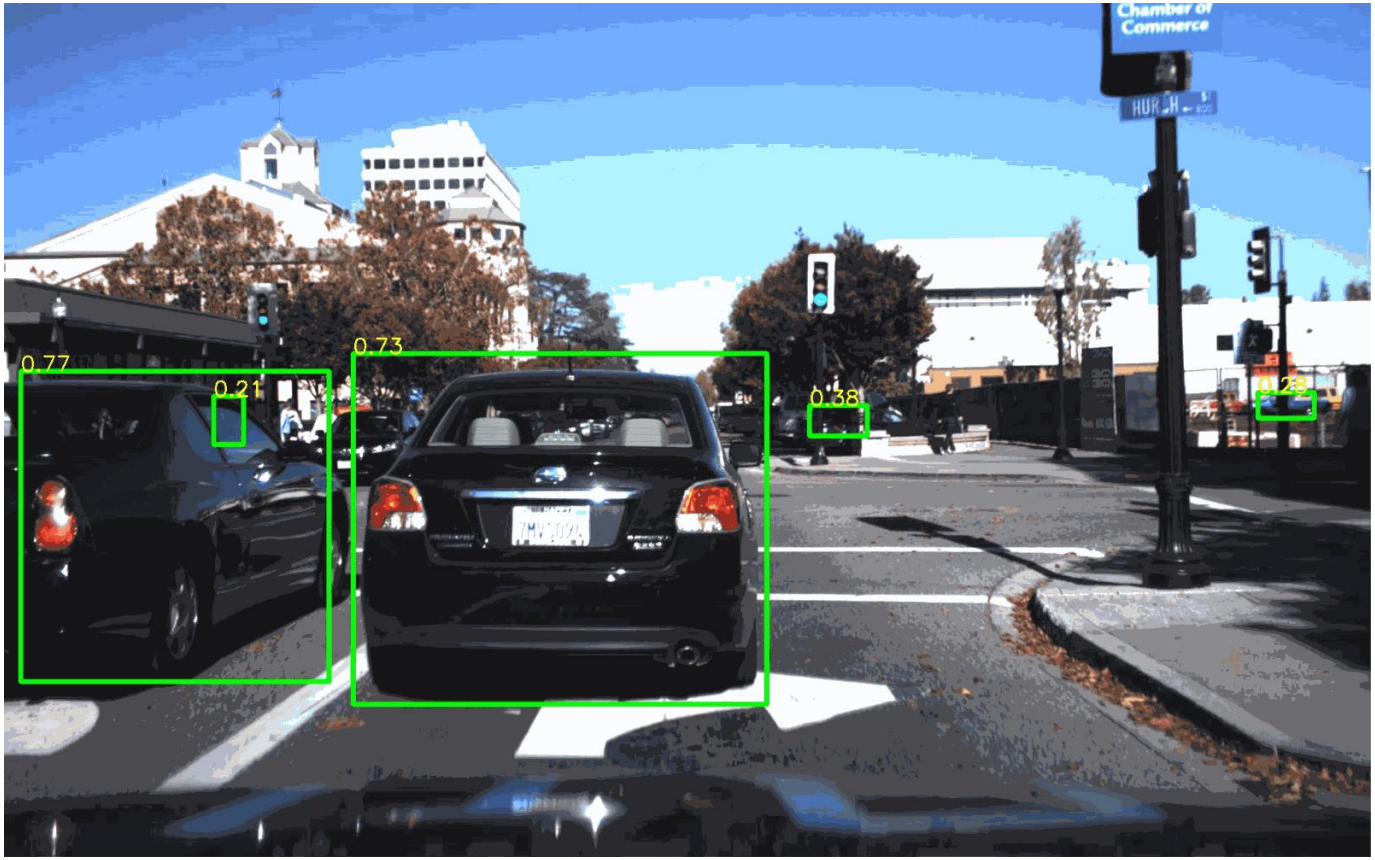


HOG detector



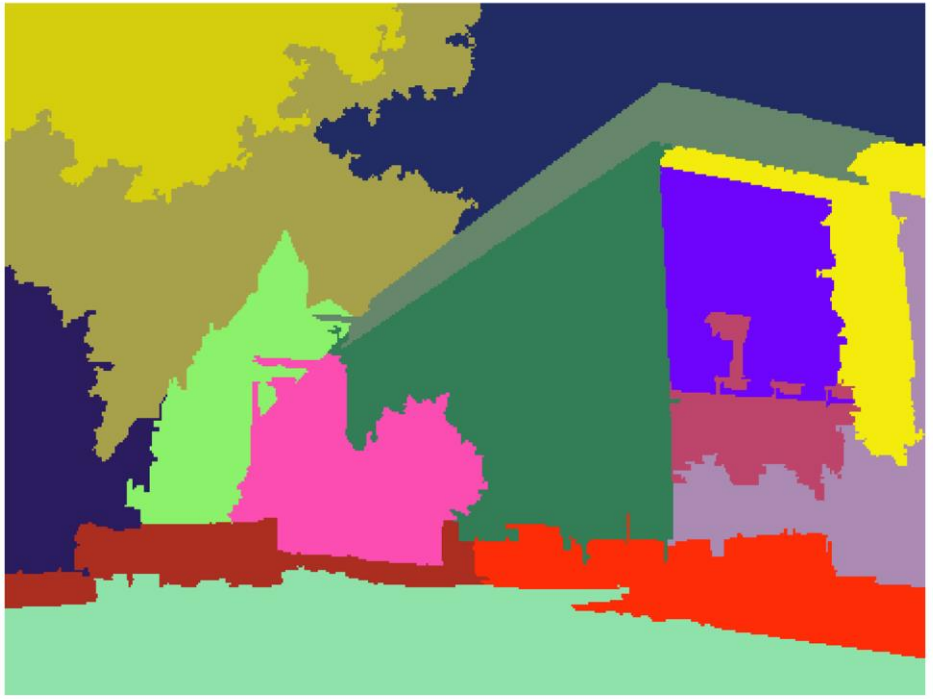
Kazemi's cascade regressor trained by gradient boosting

Object Detection



Object tracking

Segmentation

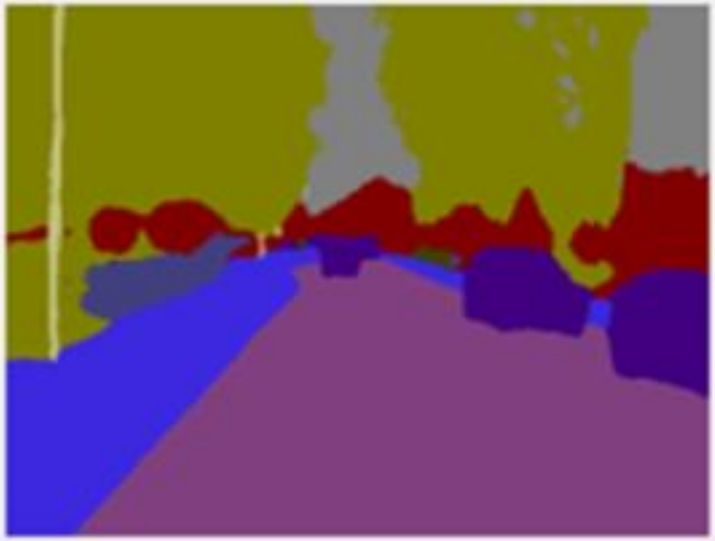


Semantic segmentation

Input

RGB Image

Output



Segmentation